AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **90(2)** (2024), Pages 110–120

Computing shortest 12-representants of labeled graphs

Asahi Takaoka

College of Information and Systems
Muroran Institute of Technology
Muroran, Hokkaido, 050–8585, Japan
takaoka@muroran-it.ac.jp

Abstract

The notion of 12-representable graphs was introduced as a variant of a well-known class of word-representable graphs. Recently, these graphs were shown to be equivalent to the complements of simple-triangle graphs. This indicates that a 12-representant of a graph (i.e., a word representing the graph) can be obtained in polynomial time if it exists. However, the obtained 12-representant is not necessarily optimal (i.e., shortest possible). This paper proposes an $O(n^2)$-time algorithm to generate a shortest 12-representant of a labeled graph if it is 12-representable, where n is the number of vertices of the graph.

1 Introduction

The theory of word-representable graphs is an active research area and provides an interesting way to connect the study of graphs with words. A graph G is *word-representable* if there is a word w over the alphabet $V(G)$ such that two letters x and y are adjacent in G if and only if a word $xyxy\cdots$ or a word $yxyx\cdots$ remains after removing all other letters from w. Such a word w is called a *word-representant* of G. The notion of word-representable graphs was introduced by Kitaev and Pyatkin [9] based on the study of the Perkins semigroup in [10]. Motivated by their relevance to various fields, such as algebra, graph theory, computer science, combinatorics on words, and scheduling [4, 8], word-representable graphs have been extensively investigated; see [6, 8] for comprehensive surveys on this topic. One of the results relevant to this paper is the NP-hardness of the recognition; see Theorem 39 of [6] or Theorem 4.2.15 of [8].

Jones et al. [5] introduced the notion of u-representable graphs as a generalization of word-representable graphs. In this context, word-representable graphs are called 11-representable graphs. Kitaev [7] showed that only two graph classes are nontrivial in the theory of u-representable graphs: 11-representable graphs and 12-representable graphs. This paper focuses on 12-representable graphs.

ISSN: 2202-3518

Let $[n] = \{1, 2, \ldots, n\}$ for a positive integer n. A labeled graph G whose labels are drawn from $[n]$ is 12-*representable* if there is a word w over $[n]$ such that each letter of $[n]$ appears at least once in w and two vertices i and j with $i < j$ are adjacent in G if and only if no i occurs before j in w. In this situation, w is said to 12-*represent* the graph G and w is called a 12-*representant* of G. For example, the graph G_1 in Figure 1(a) is 12-representable by the word $w = 8753532847616421$. An unlabeled graph G is 12-representable if there is a labeling of G which results a 12-representable labeled graph.

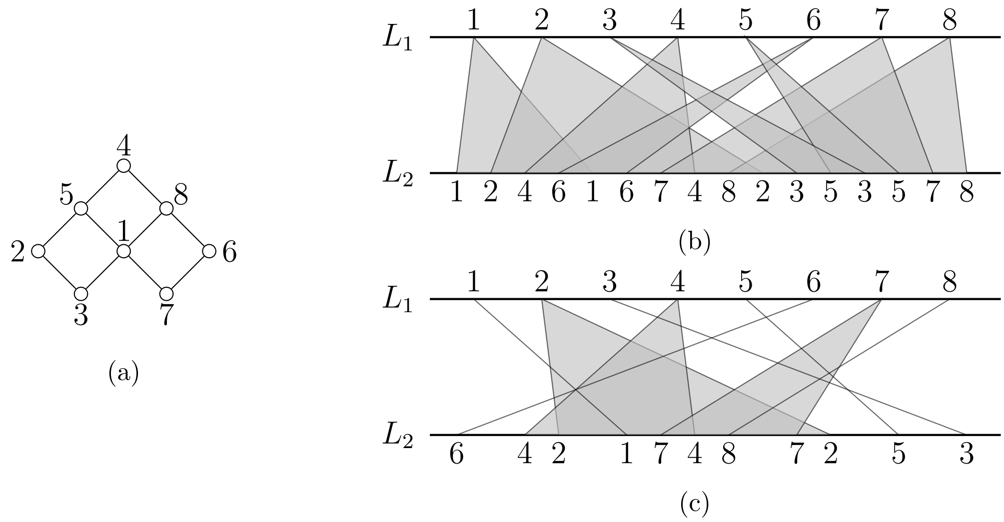

Figure 1: (a) A 12-representable graph G_1. (b) A model of the complement $\overline{G}_1$ of G_1. (c) Another model of $\overline{G}_1$. The vertices are labeled based on the points on L_1. The word $w = 8753532847616421$ obtained from the model in Figure 1(b) is a 12-representant of G_1. We can obtain another 12-representant $w' = 35278471246$ from the model in Figure 1(c). As mentioned in Example 3.5, three vertices 2, 4, and 7 are bad in G_1; hence, w' is a shortest 12-representant of G_1.

Jones et al. [5] showed that the class of 12-representable graphs is a proper subclass of comparability graphs and a proper superclass of co-interval graphs and permutation graphs. They also provided a forbidden subgraph characterization of 12-representable trees and a necessary condition for 12-representability, which turned out to be sufficient (Theorem 2.6). Chen and Kitaev [1] investigated the 12-representability of a subclass of grid graphs and presented its characterization.

The class of 12-representable graphs is equivalent to the complements of simple-triangle graphs [15]. This equivalence can be depicted as follows. Let L_1 and L_2 be two horizontal lines in the plane with L_1 above L_2. A point on L_1 and an interval on L_2 define a triangle between L_1 and L_2. A graph is a *simple-triangle graph* [2] if there is a triangle T_v for each vertex v of G such that two vertices u and v are adjacent if and only if T_u intersects T_v. The set $\{T_v\colon\ v \in V(G)\}$ of triangles is called a *model* or *representation* of G; we use the term model in this paper to avoid confusion. For example, Figure 1(b) is a model of the complement $\overline{G}_1$ of the graph G_1 in Figure 1(a). Indeed, two vertices of G_1 are adjacent if and only if the corresponding triangles do not intersect. Given a model of a simple-triangle graph, we can obtain a 12-representant of its complement by labeling each triangle based on the point on L_1 from left to right and reading the labels of endpoints on L_2 *from right to left*. For example, a 12-representant $w = 8753532847616421$ of G_1 can be obtained from the model in Figure 1(b). On the other hand, we can construct a model of a simple-triangle graph from a 12-representant of its complement since the complement admits a 12-representant in which each letter appears at most twice (Theorem 2.1).

It is worth mentioning that a simple-triangle graph admits several models, and different models can yield different 12-representants. For instance, Figure 1(c) illustrates another model of $\overline{G}_1$, which provides another 12-representant $w' = 35278471246$ of G_1. In addition, triangles can be degenerated to lines as in the model of Figure 1(c), which correspond to letters appearing only once in the 12-representant.

Since simple-triangle graphs can be recognized in $O(nm)$ time [14], the equivalence indicates that 12-representable graphs can be recognized in $O(n(\bar{m} + n))$ time [15], where n, m and $\bar{m}$ are the number of vertices, edges and non-edges of the given graph, respectively. Moreover, a 12-representant of a graph can be obtained in the same time bound if it exists.

It should be noted that the situation is somewhat different for labeled graphs, i.e., when the labeling is given. It is possible that some labeling of a graph admits a 12-representant whereas another does not (Theorem 2.6). Finding valid labeling takes $O(n(\bar{m}+n))$ time, but when a valid labeling is given, we can obtain a 12-representant in $O(n^2)$ time (Theorem 2.7).

The 12-representants obtained by the method of [15] are of length $2n$, and improving the upper bound on the length remains open [15]. This is the subject the paper deals with. The problem can also be viewed as how many triangles in the model could be degenerated to lines. The paper proposes an $O(n^2)$-time algorithm to compute a shortest 12-representant[1] of the given *labeled* graph if it is 12-representable. In particular, we show an algorithm to transform a 12-representant w of a labeled graph G to a shortest 12-representant w' of G. The algorithm is presented in Section 3. Section 2 introduces some definitions, notations and results used in this paper. Section 4 discusses the unlabeled case and poses an open question. Notably, computing a word-representant is NP-hard regardless of the labeling since the recognition is NP-hard [6, 8] and, unlike 12-representable graphs, the labeling is not important for word-representable graphs.

2 Preliminaries

Graphs. All graphs in this paper are finite, simple, and undirected. We use uv to denote the edge joining two vertices u and v. For a graph G, we use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of G, respectively. We usually denote the

[1]As will be noted in Remark 3.8, the shortest 12-representants are not necessarily unique, even for labeled graphs.

number of vertices by n. The *complement* of a graph G is the graph $\overline{G}$ such that $V(\overline{G}) = V(G)$ and $uv \in E(\overline{G})$ if and only if $uv \notin E(G)$ for any two vertices u, v of $\overline{G}$. For a graph G, a graph H is an *induced subgraph* of G if $V(H) \subseteq V(G)$ and $uv \in V(H)$ if and only if $uv \in V(G)$ for all $u, v \in V(H)$.

A *labeled graph* of a graph G is obtained from G by assigning an integer (label) to each vertex. A *labeling* of G is an assignment of labels to the vertices of G. All labels are assumed to be distinct and drawn from $[n] = \{1, 2, \ldots, n\}$. For a labeled graph, we usually denote its vertices by their labels. Unless stated otherwise, graphs are assumed to be unlabeled.

Trivial upper and lower bounds. By definition, every 12-representant contains at least one copy of each letter. Hence, n is a lower bound for the length of 12-representants. The following theorem yields the upper bound.

Theorem 2.1 ([5]). *For a* 12*-representable graph, there is a* 12*-representant in which each letter occurs at most twice.*

Hence, we have the following proposition.

Proposition 2.2. *The length of a shortest* 12*-representant of a graph is at least* n *and at most* $2n$*, where* n *is the number of vertices of the graph.*

The following theorem can also be used to obtain the lower bound.

Theorem 2.3 ([5]). *A graph is* 12*-representable by a permutation if and only if it is a permutation graph.*

Corollary 2.4. *If a graph is not a permutation graph, then the length of its* 12*-representant is at least* $n + 1$*, where* n *is the number of vertices of the graph.*

For example, applying Corollary 2.4 to the graph G_1 in Figure 1(a), we obtain the lower bound.

Example 2.5. We can see that the graph G_1 in Figure 1(a) is not a permutation graph as follows. The graph obtained from G_1 by removing the vertex 6 is isomorphic to the graph $\overline{\Gamma}_{12}[8]$ in [3, 11]. Thus, G_1 is not the complement of a comparability graph. Since any permutation graph is the complement of a comparability graph [12], the graph G_1 is not a permutation graph. Therefore, the length of every 12-representant of G_1 is at least 9.

Labeling and recognition. Labeling is important when dealing with 12-representable graphs. The following theorem indicates that not all labelings of a 12-representable graph are 12-representable.

Theorem 2.6 ([15]). *A labeled graph* G *is* 12*-representable if and only if* G *contains no induced subgraph* H *such that* $\mathrm{red}(H)$ *is equal to one of* I_3*,* J_4*, or* Q_4 *in Figure 2, where* $\mathrm{red}(H)$ *denotes the* reduced form *of* H*, i.e., the labeled graph obtained by relabeling* H *so that the* i*-th smallest label is replaced by* i*.*

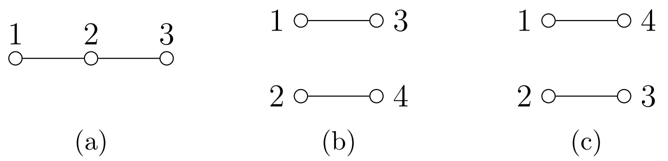

Figure 2: The labeled graphs (a) I_3, (b) J_4, and (c) Q_4.

It should be noted that the necessity in Theorem 2.6 was first presented by Jones et al. [5]. We call a labeling *valid* if its resulting graph does not contain an induced subgraph isomorphic to I_3, J_4, or Q_4 in the reduced form.

As mentioned in the introduction, 12-representable graphs are exactly the complements of simple-triangle graphs [15]. It follows that 12-representable graphs can be recognized in $O(n(\bar{m}+n))$ time, where $\bar{m}$ is the number of non-edges of the given graph. However, when a valid labeling is given, we can obtain its 12-representant in $O(n^2)$ time.

Theorem 2.7 ([15]). *From a valid labeling of a* 12*-representable graph* G*, a* 12*-representant of* G *can be obtained in* $O(n^2)$ *time without relabeling of* G*.*

In more detail, given a valid labeling of a 12-representable graph G, we can obtain a model of its complement $\overline{G}$ in $O(n^2)$ time using the algorithm in [13]. A 12-representant of G can be obtained from the model described in the introduction, and it will contain at most two occurrences of each letter. Hence, we have the following theorem.

Theorem 2.8. *Given a graph with valid labeling, its* 12*-representant in which each letter occurs at most twice can be obtained in* $O(n^2)$ *time.*

Remark 2.9. In [13], the author showed that, given a labeled graph G, a model of its complement $\overline{G}$ can be constructed in $O(n^2)$ time, provided that the labeling of G is valid. However, the case where the labeling of G is not valid (i.e., the labeled graph G is not 12-representable) was not discussed. The question of whether a given labeled graph can be checked to be non-12-representable in $O(n^2)$ time remains unresolved.

3 Algorithm

We first improve the lower bound for the length of 12-representants.

Definition 3.1. Let G be a labeled graph. We refer to a vertex b of G as a *bad vertex* if there exist two vertices a and c with $a < b < c$ such that $ab, bc \notin E(G)$ and $ac \in E(G)$. We call a vertex *good* if it is not a bad vertex.

Proposition 3.2. *Let* G *be a* 12*-representable labeled graph. Each bad vertex must occur twice in every* 12*-representant of* G*.*

Proof. Let w be a 12-representant of G, and let a, b, and c be three vertices with $a < b < c$ such that $ab, bc \notin E(G)$ and $ac \in E(G)$. Since $ac \in E(G)$, every copy of c occurs before the first occurrence of a in w. Then, $ab, bc \notin E(G)$ implies that b occurs after some a and before some c in w. □

Proposition 3.2 leads to the following lower bound.

Lemma 3.3. *Let G be a labeled graph. The length of every 12-representant of G is at least $n + b$, where n and b are the number of vertices and bad vertices of G, respectively.*

In the rest of this section, we show that the length of shortest 12-representants of a labeled graph is exactly $n + b$. Suppose that we are given a 12-representable graph G and its 12-representant w. By Theorem 2.1, we can assume that each letter occurs at most twice in w. Proposition 3.2 states that all bad vertices occur twice in w. If the length of w is larger than $n + b$, then some good vertices occur twice in w. Therefore, we propose an algorithm to transform w to another 12-representant of G in which no good vertices occur twice.

The following is a key observation.

Proposition 3.4. *Let G be a labeled graph with a 12-representant w. Suppose that a letter i occurs twice in w.*

(a) *Let j be a letter just after the first occurrence of i, i.e., $w = W_1ijW_2iW_3$, where W_1, W_2, and W_3 are subwords of w. If $j < i$ then the word $w' = W_1jiW_2iW_3$ is a 12-representant of G.*

(b) *Let j be a letter just before the second occurrence of i, i.e., $w = W_1iW_2jiW_3$, where W_1, W_2, and W_3 are subwords of w. If $j > i$ then the word $w' = W_1iW_2ijW_3$ is a 12-representant of G.*

(c) *If two occurrences of i are consecutive in w, we can remove one occurrence. In other words, if $w = W_1iiW_2$, where W_1 and W_2 are subwords of w, then the word $w' = W_1iW_2$ is a 12-representant of G.*

Proof. (a) Since j occurs before the second occurrence of i in w, we have $ij \notin E(G)$. The letter j still occurs before i in w', and hence, w' is a 12-representant of G. (b) This can be proved similarly to (a). (c) Trivial. □

Proposition 3.4 leads to Algorithm 1, which generates a 12-representant in which no good vertices occur twice. The algorithm works under the assumption that each letter occurs at most twice in the input. The case where some letters occur more than twice will be discussed in Remark 3.9.

Before proving the correctness of Algorithm 1, we see how the algorithm works.

Example 3.5. Recall that the word $w = 8753532847616421$ is a 12-representant of the graph G_1 in Figure 1(a). Applying Algorithm 1 to w, we obtain the 12-representant $w' = 35278471246$. The operation is illustrated in Figure 3. Since three vertices 2, 4, and 7 are bad in G_1, the word w' is the shortest.

Now, we prove the correctness of Algorithm 1.

Theorem 3.6. *Algorithm 1 computes a shortest 12-representant of the labeled graph 12-represented by the input.*

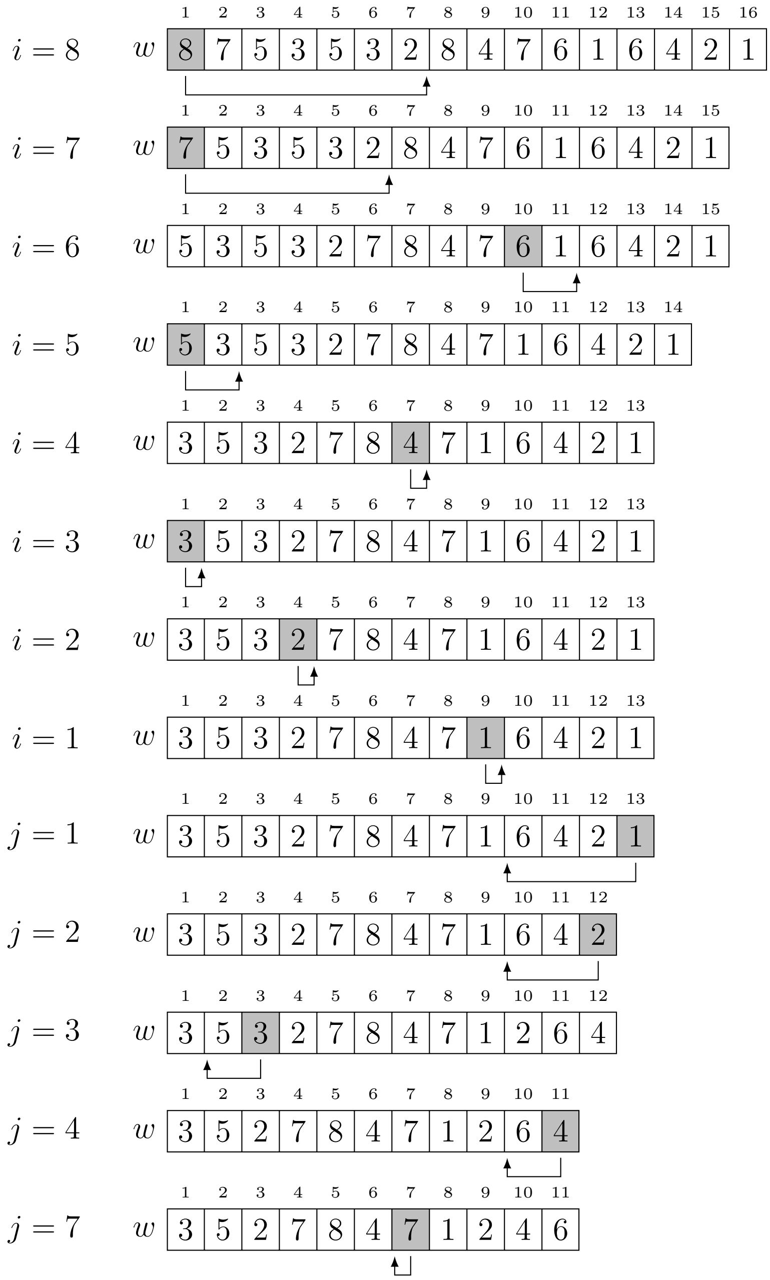

Figure 3: The operation of Algorithm 1 on the word $w = 8753532847616421$. Each letter is stored in the box whose position appears above. In each iteration, the letter in the shaded box moves, and the arrow denotes the move. The first eight lines illustrate the loops in lines 1–7 while the remaining lines illustrate those in lines 8–14. We omitted the cases $j = 5$, 6, and 8 because these letters do not appear twice in w at the time.

Algorithm 1: Computing a shortest 12-representant of a labeled graph

Input: A 12-representant $w = w_1 w_2 \dots w_\ell$ of a labeled graph G.
Output: A shortest 12-representant of G.

`// We assume that each letter occurs at most twice in` w.

1 **for** $i \leftarrow n$ **downto** 1 **do**
2   **if** *i occurs twice in w* **then**
3   Set p to the position of the first occurrence of i in w;
4   **while** $w_p > w_{p+1}$ **do** swap w_p and w_{p+1}; $p \leftarrow p + 1$;
5   **if** $w_p = w_{p+1}$ **then** remove w_p from w;
6   **end**
7 **end**
8 **for** $j \leftarrow 1$ **to** n **do**
9   **if** *j occurs twice in w* **then**
10   Set q to the position of the second occurrence of j in w;
11   **while** $w_q < w_{q-1}$ **do** swap w_q and w_{q-1}; $q \leftarrow q - 1$;
12   **if** $w_q = w_{q-1}$ **then** remove w_q from w;
13   **end**
14 **end**
15 **return** w.

Proof. Let G and w denote the graph and the input, respectively. Proposition 3.4 ensures that the output is still a 12-representant of G. By Proposition 3.2, bad vertices occur twice in any 12-representant. Hence, it suffices to prove that no good vertices occur twice in the output.

Let j be a good vertex of G. Suppose to the contrary that j occurs twice in w at the end of the jth loop in lines 8–14.

Claim 1. Let i be a letter with $i \geq j$ appearing twice in w at the end of the jth loop in lines 8–14. If p is the first position of i, then $w_p < w_{p+1}$.

Proof of Claim 1. According to Algorithm 1, we can see $w_{p'} < w_{p'+1}$ at the end of the $(n-i+1)$th loop in lines 1–7, where p' is the first position of i at the time. After the $(n-i+1)$th loop, only letters smaller than i move forward. Hence, $w_{p''} < w_{p''+1}$ at the beginning of the first loop in lines 8–14, where p'' is the first position of i at the time. Before the end of the jth loop in lines 8–14, only letters smaller than or equal to i move backward. Thus, the claim holds. □

Claim 2. Let k be a letter with $k \leq j$ appearing twice in w at the end of the jth loop in lines 8–14. If q is the second position of k, then $w_q > w_{q-1}$.

Proof of Claim 2. According to Algorithm 1, we can see $w_{q'} > w_{q'-1}$ at the end of the kth loop in lines 8–14, where q' is the second position of k at the time. After the kth loop, only letters larger than k move backward. Thus, the claim holds. □

Let p and q be the first and second positions of j, respectively, at the end of the jth loop in lines 8–14. We have $w_{p+1} > w_p$ from Claim 1 and $w_{q-1} < w_q$ from Claim 2. Let $i_1 = w_{p+1}$ and $k_1 = w_{q-1}$, i.e., $w = W_1 j i_1 W_2 k_1 j W_3$, where W_1, W_2, and W_3 are subwords of w. We have $k_1 < j < i_1$ and $k_1 j, j i_1 \notin E(G)$. Since j is a good vertex, $k_1 i_1 \notin E(G)$. Hence, some k_1 occurs before some i_1 in w. It follows that another i_1 occurs after w_{p+1} or another k_1 occurs before w_{q-1}.

Suppose that i_1 occurs after w_{p+1}. Claim 1 indicates $w_{p+2} > w_{p+1}$. Let $i_2 = w_{p+2}$, i.e., $w = W_1 j i_1 i_2 W_4 k_1 j W_3$, where W_4 is a subword of w. We have $k_1 < j < i_2$ and $k_1 j, j i_2 \notin E(G)$. Since j is a good vertex, $k_1 i_2 \notin E(G)$. Hence, some k_1 occurs before some i_2 in w. It follows that another i_2 occurs after w_{p+2} or another k_1 occurs before w_{q-1}.

On the other hand, suppose that k_1 occurs before w_{q-1}. Claim 2 indicates $w_{q-2} < w_{q-1}$. Let $k_2 = w_{q-2}$, i.e., $w = W_1 j i_1 W_5 k_2 k_1 j W_3$, where W_5 is a subword of w. We have $k_2 < j < i_1$ and $k_2 j, j i_1 \notin E(G)$. Since j is a good vertex, $k_2 i_1 \notin E(G)$. Hence, some k_2 occurs before some i_1 in w. It follows that another i_1 occurs after w_{p+1} or another k_2 occurs before w_{q-2}.

Continuing in this way, we obtain an infinite sequence $w_p < w_{p+1} < w_{p+2} < \cdots$ or $w_q > w_{q-1} > w_{q-2} > \cdots$, which is a contradiction. □

From Theorem 3.6, we have the main theorem.

Theorem 3.7. *The length of a shortest 12-representant of a labeled graph is $n + b$, where n and b are the number of vertices and bad vertices of the graph, respectively. A shortest 12-representant of a labeled graph can be obtained in $O(n^2)$ time if it is 12-representable.*

Proof. The proof of Theorem 3.6 indicates the first statement. Theorem 2.8 states that a 12-representant of a labeled graph in which each letter occurs at most twice can be obtained in $O(n^2)$ time if it exists. It is obvious that Algorithm 1 takes $O(n^2)$ time. Thus, the second statement holds. □

Remark 3.8. The idea behind Algorithm 1 is to apply the operations in Proposition 3.4 to shorten the given word 12-representing the graph. Different ways of applying these operations may result in different shortest 12-representants. To illustrate this, consider the word $w = 8753532847616421$ used in Example 3.5, which is a 12-representant of the graph G_1 in Figure 1(a). Applying the procedure in lines 3–5 of Algorithm 1 only when $i = 8$, 6, and 5, and in lines 10–12 only when $j = 1$ and 3, we obtain the word $w'' = 73528471642$, which is a 12-representant of G_1 different from w' in Example 3.5.

Remark 3.9. As mentioned before, we assume the input to Algorithm 1 contains at most two occurrences of each letter. This does not lose generality because when some letters occur more than twice, we can remove occurrences of the letters except for the first and last occurrences, which results in a 12-representant of the same graph; see Theorem 8 of [5]. It should be noted that the procedure takes $O(\ell)$ time, where ℓ is the length of the input, and therefore, the time complexity is no longer $O(n^2)$ if $\ell = \omega(n^2)$ (i.e., ℓ is asymptotically strictly larger than n^2).

4 Concluding remarks

This paper proposes an $O(n^2)$-time algorithm to transform a 12-representant w of a labeled graph G to a shortest 12-representant w' of G, where n is the number of vertices of G. This indicates that shortest 12-representants of labeled graphs can be obtained in $O(n^2)$ time if it exists. The natural next step is to study the unlabeled case, i.e., the problem of finding a shortest 12-representant of the given unlabeled graph.

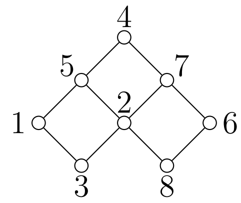

Figure 4: Another labeling G_2 of the graph G_1 in Figure 1(a). As shown in Theorem 2.18 of [1], the word $w = 351748246$ is a 12-representant of G_2.

For 12-representability, labeling matters from the existential point of view (Theorem 2.6). The labeling also matters to find a shortest 12-representant of an unlabeled graph. In other words, the shortest 12-representant of some labeling of a graph G can be shorter than that of another labeling of G. For example, as shown in Example 3.5, the shortest 12-representant of the graph G_1 in Figure 1(a) is of length $11 = n + 3$. However, if we relabel G_1 as G_2 in Figure 4, then we obtain a 12-representant $w = 351748246$ of length $9 = n + 1$, as shown in Theorem 2.18 of [1]; only the vertex 4 is bad in G_2. We can see from Example 2.5 that w is a shortest 12-representant of the unlabeled graph. Therefore, we conclude this paper by posing the following open question.

Problem 4.1. Given an unlabeled graph G, can we compute a valid labeling of G minimizing the number of bad vertices in polynomial time?

Acknowledgments

The author is grateful to the reviewers for their careful reading and helpful comments. This work was supported by JSPS KAKENHI Grant Number JP23K03191.

References

[1] J. N. Chen and S. Kitaev, On the 12-representability of induced subgraphs of a grid graph, *Discuss. Math. Graph Theory* **42**(2) (2022), 383–403; `doi:10.7151/dmgt.2263`.

[2] D. G. Corneil and P. A. Kamula, Extensions of permutation and interval graphs, *Congr. Numer.* **58** (1987), 267–275.

[3] T. Gallai, Transitiv orientierbare graphen, *Acta Math. Acad. Sci. Hungar.* **18**(1) (1967), 25–66; doi:10.1007/BF02020961.

[4] M. M. Halldórsson, S. Kitaev and A. V. Pyatkin, Semi-transitive orientations and word-representable graphs, *Discret. Appl. Math.* **201** (2016), 164–171; doi:10.1016/j.dam.2015.07.033.

[5] M. Jones, S. Kitaev, A. Pyatkin and J. Remmel, Representing graphs via pattern avoiding words, *Electron. J. Combin.* **22**(2) (2015), #P2.53 (20 pp); doi:10.37236/4946.

[6] S. Kitaev, A comprehensive introduction to the theory of word-representable graphs, in: *Developments in Language Theory, DLT 2017, Lec. Notes in Comput. Sci.* **10396**, É. Charlier, J. Leroy and M. Rigo (eds), Springer (2017), 36–67; doi:10.1007/978-3-319-62809-7_2.

[7] S. Kitaev, Existence of u-representation of graphs, *J. Graph Theory* **85**(3) (2017), 661–668; doi:10.1002/jgt.22097.

[8] S. Kitaev and V. Lozin, *Words and Graphs*, Monographs in Theoretical Computer Science, An EATCS Series, Springer, 2015; doi:10.1007/978-3-319-25859-1.

[9] S. Kitaev and A. V. Pyatkin, On representable graphs, *J. Autom. Lang. Comb.* **13**(1) (2008), 45–54; doi:10.25596/jalc-2008-045.

[10] S. Kitaev and S. Seif, Word problem of the Perkins semigroup via directed acyclic graphs, *Order* **25**(3) (2008), 177–194; doi:10.1007/S11083-008-9083-7.

[11] F. Maffray and M. Preissmann, A Translation of Tibor Gallai's Paper: 'Transitiv Orientierbare Graphen', in: *Perfect Graphs*, J. L. Ramírez-Alfonsín and B. A. Reed (eds), Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley (2001), 25–66.

[12] A. Pnueli, A. Lempel and S. Even, Transitive orientation of graphs and identification of permutation graphs, *Canad. J. Math.* **23(1)** (1971), 160–175; doi:10.4153/CJM-1971-016-5.

[13] A. Takaoka, A vertex ordering characterization of simple-triangle graphs, *Discrete Math.* **341(12)** (2018), 3281–3287; doi:10.1016/j.disc.2018.08.009.

[14] A. Takaoka, A recognition algorithm for simple-triangle graphs, *Discret. Appl. Math.* **282** (2020), 196–207; doi:10.1016/j.dam.2019.11.009.

[15] A. Takaoka, Graph classes equivalent to 12-representable graphs, *Discuss. Math. Graph Theory* **44(3)** (2024), 1023–1035; doi:10.7151/dmgt.2486.

(Received 19 May 2023; revised 1 July 2024, 8 Sep 2024)